Lecture 6 - Physics Subsystem

"Part 1. Particle Physics" - Ian Millington

- Vector maths
- Game object interaction
- Physics rules: mostly fake
- Approximations / performance

Last Week Recap

- ECS and game world

Objective

- Mainly to give you a bit of taste for physics breadth in game dev
- Don't worry if you don't understand *all* of it: awareness is the key
- The engine takes care of most things for us . . . oh, wait a minute!

Game Dev Maths

- Fundamental to the field
- Ranges from basic to very complex
- Important to know the difference between what, why and how

2D Maths

- Luckily, not very complex
- Transitive skills from / to UI design
- Examples to consider: Arx Fatalis, Sony PS3, Valve games

Point

- Ordered pair: (x, y)
- Location / position in space
- Extendible to n dimensions
- Crucial to most games

Game World vs Screen

- Positive Y axis
- Not always 1 to 1 mapping
- Visible area (viewport into *infinite* world)

Point (Use Case)

- Distance to things
- Relative position to things

Activity

1. Construct a 4D point, where 4th component is time.
2. Can you design a game where a 4D point is used in gameplay?

Vector

- Ordered pair: (x, y) - yes, same as point
- Displacement with direction and length (magnitude)
- Describes velocity, acceleration, direction, etc.
- Extendible to n dimensions

Vector (Use Case)

- Distance to things (recall point)
- Direction of an object
- Force / speed via length
- Angle representation

Activity

Identify which part of ArxLibertatis code handles random vectors.

Transformations (Translate)

- Means move
- Apply over time to create a moving animation
- Apply with acceleration to simulate movement

Transformations (Scale)

- Means growing or shrinking
- Apply over time to create an animation
- Changes object dimensions (note for collisions)

Transformations (Rotate)

- Means changing angle
- Apply over time to create a spinning animation
- Changes occupying space (note for collisions)

Activity

1. Create (on paper or in code) a moving animation using point and vector concepts.
2. Apply acceleration
3. Apply gravity

Interpolations (Demo)

Changing the rate of change. Only a demo in this lecture, the concepts will be covered later.

Physics - Representing Game Objects

- Objects have dimensions, e.g. width, height, depth
- Objects have an *anchor* (position)

Game Physics

- Speed matters (functions get called many times in one frame)
- Need not be precise, just good enough
- Can utilise game specifics (e.g. platformer, tile based)

Collision Detection Phases

- Broad phase
- Narrow phase

AABB (Axis Aligned Bounding Box)

- Very fast
- Very easy to implement
- Extends to n dimensions
- Works in cases without rotations

Activity

Identify which part of the FXGL code base performs AABB.

High-level Implementation

Suppose we have `n` entities, how can we implement collision detection for all of them?

SAT (Separating Axis Theorem)

- Slower
- Easy to implement
- Extends to n dimensions
- Works in cases with rotations

Activity

Identify which part of the FXGL code base performs SAT.

Grids

- Fast
- Easy to implement
- Extends to n dimensions
- Works in tile / grid based cases, as broad phase

Space Partioning (e.g. Quadtree)

- Speed depends on use case
- Harder to get “right”
- Extends to n dimensions
- Works only as broad phase

Pixel-level / Bitmap

- Very slow
- Relatively easy to implement
- Works only in 2D
- Works in cases with rotations

Continuous / Sweeping

- Slower
- Harder to get “right”
- Extends to n dimensions
- Mainly for fast moving objects or simulations

Special Cases

- Circle-circle collision
- Polygon clipping

Activity

1. Add two entities to your game demo and make them collide.
2. Handle collision by printing “Collision” to console or similar.

High-level Usage

Old school

```
if (obj1.collidesWith(obj2)) {
    // do smth
}
```

Modern

```
addHandler(type1, type2, (obj1, obj2) -> {
    // do smth
})
```

Conclusion

- Physics and maths are very important for games
- Don't have to get exactly right
- Cheap and cheerful algorithms exist

Further Reading

- Good SAT tutorial